Introduction

The following recommended naming standards guidelines were originally developed by Simon Kingston (WASO-I&M) and Angie Southwould (AKSO) after the 2002 NPS I&M Data Manager's meeting and distributed for comment and review in October of the same year. They were intended to support the development of I&M MS Access database applications in general. The recommendations were reviewed and revised by Chris Dietrich (WASO-I&M) after the 2004 Data Manager's meeting with the intent of adopting them as the standard naming convention for Natural Resource Database Template (NRDT) applications (see National Park Service Natural Resource Database Template (NRDT) Data Standards Submission and Approval Process, http://www1.nrintra.nps.gov/im/apps/BOFAT). While not everyone may embrace specific recommendations of a particular naming convention, most would agree that adopting a convention and applying it consistently are important for data sharing and application maintenance and support. Because many NRDT applications are designed for long-term (more than 5 years) use, I&M strongly recommends that developers, including NPS staff, contractors and partners, follow the naming standards outlined in this document to enhance continuing application support and to maximize standardization among NRDT applications.

Rule #1	Recommended Implementation
Prefix table names	• Prefix each table with the appropriate category abbreviation. • There are three categories of tables. A data table is most common and contains data collected in the field. A lookup table contains a list of valid values that references some other field in the database. A cross-reference (or linking) table is the table created to accurately depict a many-to-many relationship; this table references the two parent tables and contains their primary keys. • To help distinguish a table type, prefix the table name with one of the following: • Data Table = tbl_ • Lookup Table = tlu_
	Examples of Recommended Design
	tbl_Locations tbl_Events tlu_Parks tlu_Observers xref_Location_Event xref_Event_Bird_Observation

Rule #2	Recommended Implementation

Prefix objects	**Prefix each database object with the appropriate abbreviation.**
	The following list contains some recommended prefixes for other database objects:
	• Form
	  • Main Form = frm_
	  • Query (for criteria) = frm_qry_
	  • List (read-only grid) = frm_lst_
	  • Edit (for data entry) = frm_edt_
	  • Sub Form = frm_sub_
	• Query
	  • Select = qry_
	  • Insert = qry_ins_
	  • Update = qry_upd_
	  • Delete = qry_del_
	  • Append = qry_app_
	  • Make-Table = qry_mak_
	  • Cross-Tab = qry_xtab_
	• Report
	  • Main Report = rpt_
	  • Sub Report = rpt_sub_
	• Macro = mac_
	• Module = mod_
	• Index = idx_

Rule #3	Recommended Implementation	
Avoid spaces	**Do not use spaces within a file, table or field name.** Many users include spaces in file, table and field names for easy readability. Instead, use an underscore between words (see **Separate words** rule – Rule #4). Access is one of the only databases to even allow the use of spaces within names, so other databases, development tools, and analysis tools often have difficulty using these fields without custom workarounds added by the programmer or user.	
	Examples of Design	**Examples of Better/Recommended Design**
	Location ID	Location_ID
	Park Code	Park_Code
	Project Code	Project_Code

Rule #4	Recommended Implementation	
Separate words	**Use underscores to separate words within a single table or field name.** Many users rely on mixed case table and field names for easy readability. Rather than relying upon the way the name is typed, a better choice is to separate words with an underscore (_). The name then resembles written text and words within the name are easily identified. Many analysis tools ignore case. When data is imported from a database, the tool converts field names to all upper or all lower case. By using underscores to separate words, the fields maintain readability. However, keep in mind that some databases are case-sensitive. This means that any user must type the names exactly as they have been created to get the expected results.	
	Examples of Design	**Examples of Better/Recommended Design**
	EventID	Event_ID
	StartDate	Start_Date
	EndDate	End_Date
	AlphaCD	Alpha_CD

Rule #5	Recommended Implementation	
Specific names	**Choose a name that accurately identifies the data to be stored in a table or field.** Strive to create names that accurately define the data stored within. If a name is too vague, users must rely upon supplemental documentation for definitions. Also, users may enter data that the field was not intended to store.	
	Examples of Design	**Examples of Better/Recommended Design**
	Habitat	Viereck_Class_Code
	Percent_Cover	Tree_Cover_Percent
	Tree_Size	Tree_DBH_cm

Rule #6	Recommended Implementation	
Consistent case	**Use mixed case text within a table or field name.** All uppercase words are especially straining to the human eye. Mixed case text presents a readable format that is more easily and quickly read. If data will be ported to na enterprise-level database system (i.e., Oracle) consider using uppercase object and field names. SQL Server will honor mixed case object and field names.	
	Examples of Design	**Examples of Better/Recommended Design**
	SPECIES_CODE	Species_Code
	SPECIES_COUNT	Species_Count
	behavior_code	Behavior_Code

Rule #7	Recommended Implementation	
Avoid special characters	**Do not use special characters in a table or field name.** Allowable characters include A-Z, a-z, 0-9, and _ (underscore to separate words). Additionally, a name should never begin with a number. Do not use dashes in table or field names. Access is one of the only databases to even allow the use of special characters within names, so other databases, development tools, and analysis tools often have difficulty using these fields without custom workarounds added by the programmer or user.	
	Examples of Design	**Examples of Better/Recommended Design**
	Project Park/Region	Project_Park_Or_Region
	Project$	Project_Cost
	ProjLead_Phone#	Proj_Lead_Phone_Num
	Project Complete?	Is_Project_Complete

Rule #8	Recommended Implementation	
Avoid unknown abbreviations	**Avoid abbreviations unless necessary due to field length.** If an abbreviation is needed, make every attempt to use one that is known within the organization or one that can be easily deciphered. Avoid an abbreviation that is a word itself or has multiple meanings. Some older flat file systems limit table (file) and field names to 8 or 10 characters, respectively. Most relational databases have a limit of approximately 30 characters. Since Access and Oracle are the NPS standards and they do not impose these limits, mandating the use of terse names and extensive abbreviating for all tables and fields is not necessary. However, it is strongly recommended that spatial data or attribute data which could be imported into GIS or other PC database software (ArcView, dBase, etc.) use 8 character maximum table and 10 character maximum field names. Keep in mind that in cases where data is exported into a DBF, table and field names longer than 8 or 10 characters, respectively, will be truncated upon import, potentially sacrificing information by resulting in duplicative or unclear naming. In other cases, take advantage of additional characters to eliminate the ambiguity of table and field names.	
	Examples of Design	**Examples of Better/Recommended Design**
	SmpTrnID	Sample_Transect_ID
	Spp_Cd	Spp_Code
	VeStCnt	Vertical_Strata_Count
	GeoLocateID	Site_ID

Rule #9	Recommended Implementation	
Limit length	**Limit the length of table and field names to approximately 20 characters maximum (also see Rule #8).** This limit is set more for practicality than any other reason. Most database servers do have a maximum limit of approximately 30 characters, though. Shorter names can be typed more quickly and are easier to remember. Also, longer names can sometimes extend past the width of set drop down lists, so only the first part of a name is visible. If data will be used in GIS attribute tables for shapefiles (i.e., dBase format), consider limiting field names to 10 characters. Find an optimal field length where the name is not too tedious, but the name still clearly represents the data stored in it.	
	Examples of Design	**Examples of Better/Recommended Design**
	Water_Quality_Evaluation_Code	H2O_Quality_Eval_Code
	Geomorphic_Disturbance_Description	Geomorphic_Disturb_Desc
	Area_Average_Azimuth_ Magnetic	Area_Avg_Azimuth_Mag

Rule #10	Recommended Implementation	
Primary or Foreign Key Identification	**Use a standard suffix for primary and foreign keys.** Identify the primary keys with a _ID suffix and foreign keys with the _IDF suffix.	
	Examples of Design	**Examples of Better/Recommended Design**
	SPECIES_KEY	Species_ID
	EventFKey	Event_IDF

Rule #11	Recommended Implementation	
Single value	**Choose a field to contain a single value.** Data entry, validation and retrieval are more difficult when a single field contains multiple, independent values.	
	Examples of Design	**Examples of Better/Recommended Design**
	Full_Name	First_Name, Last_Name
	City_State_Zip	City_Name, State_Code, Zip_Code

Rule #12	Recommended Implementation	
Avoid storing calculations	**Choose a field to be independent of all other field values.** Rather than storing a calculated value in the database, a better choice is to store in independent operands and perform calculations dynamically for display within queries, forms, or reports. Stored calculations run the risk of not being updated when one of the individual elements changes. However, in certain instances, a trade-off in efficiency vs. storage space or security concerns may necessitate storage of calculated values.	
	Examples of Design	**Examples of Better/Recommended Design**
	Plot_Area_m2	Plot_Width_m, Plot_Length_m

Rule #13	Recommended Implementation	
Singularize names	**Choose the singular noun or noun_adjective form for a field name.** Where applicable, try to use singular Noun_Adjective format rather than plural Noun_Adjective or Adjective_Noun structures.	
	Examples of Design	**Examples of Better/Recommended Design**
	Life_Stages	Life_Stage
	Scientific_Species_Name	Species_Name_Scientific

Rule #14	Recommended Implementation
Avoid reserved words	**Avoid a table or field name that is a word reserved for use by the database server.** Each database server and development environment has a set of reserved words that should not be used as table or field names. Access, in particular, will allow the creation of a field that is a reserved word. It will often not cause problems until a later time, during the creation and execution of queries, forms, or reports.

Microsoft Jet Database Engine SQL Reserved Words (Source: Microsoft Access 97 help file)

The following list includes all words reserved by the Microsoft Jet database engine for use in SQL statements. The words in the list that aren't in all uppercase letters are also reserved by other applications. Words followed by an asterisk (*) are reserved but currently have no meaning in the context of a Microsoft Jet SQL statement (for example, Level and TableID).

A
ADDALL
Alphanumeric — See TEXTALTER
And
ANY
AS
ASC
AUTOINCREMENT — See COUNTER
Avg

B-C
Between
BINARY
BIT
BOOLEAN — See BIT
BY
BYTE
CHAR, CHARACTER — See TEXT
COLUMN
CONSTRAINT
Count
COUNTER
CREATE
CURRENCY

D
DATABASEDATE — See DATETIME
DATETIME
DELETE
DESC
DISALLOW
DISTINCT
DISTINCTROW
DOUBLE
DROP

E-H
Eqv
EXISTS
FLOAT, FLOAT8 — See DOUBLE
FLOAT4 — See SINGLE
FOREIGN
FROM
GENERAL — See LONGBINARY
GROUP
GUID
HAVING

I
IEEEDOUBLE — See DOUBLE
IEEESINGLE — See SINGLE
IGNORE
Imp
In
IN
INDEX
INNER
INSERTINT, INTEGER, INTEGER4 — See LONG
INTEGER1 — See BYTE
INTEGER2 — See SHORT
INTO
Is

J-M
JOIN
KEY
LEFT
Level*
Like
LOGICAL, LOGICAL1 — See BITLONG
LONGBINARY
LONGTEXT
Max
MEMO — See LONGTEXT
Min
Mod
MONEY — See CURRENCY

N-P
Not
NULL
NUMBER — See DOUBLE
NUMERIC — See DOUBLE
OLE
OBJECT — See LONGBINARY
On
OPTION
Or
ORDER
Outer*
OWNERACCESS
PARAMETERS
PERCENT
PIVOT
PRIMARY
PROCEDURE

Q-S
REAL — See SINGLE
REFERENCES
RIGHT
SELECT
SET
SHORT
SINGLE
SMALLINT — See SHORT
SOME
StDev
StDevP
STRING — See TEXTSum

T-Z
TABLE
TableID*
TEXT TIME — See DATETIME
TIMESTAMP — See DATETIME
TOPTRANSFORM
UNION
UNIQUE
UPDATE
VALUE
VALUES
Var
VARBINARY — See BINARY
VARCHAR — See TEXT
VarP
WHERE
WITH
Xor
YESNO — See BIT

Microsoft® SQL Server™ 2000 Reserved Keywords (Source: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/tsqlref/ts_ra-rz_9oj7.asp. Date accessed: 6/22/2004).

Microsoft® SQL Server™ 2000 uses reserved keywords for defining, manipulating, and accessing databases. Reserved keywords are part of the grammar of the Transact-SQL language used by SQL Server to parse and understand Transact-SQL statements and batches. Although it is syntactically possible to use SQL Server reserved keywords as identifiers and object names in Transact-SQL scripts, this can be done only using delimited identifiers. In addition, the SQL-92 standard defines a list of reserved keywords. Avoid using SQL-92 reserved keywords for object names and identifiers. The ODBC reserved keyword list (shown below) is the same as the SQL-92 reserved keyword list. **Note:** The SQL-92 reserved keywords list sometimes can be more restrictive than SQL Server and at other times less restrictive. For example, the SQL-92 reserved keywords list contains INT, which SQL Server does not need to distinguish as a reserved keyword. Transact-SQL reserved keywords can be used as identifiers or names of databases or database objects, such as tables, columns, views, and so on. Use either quoted identifiers or delimited identifiers. The use of reserved keywords as the names of variables and stored procedure parameters is not restricted.

ADD	CONTAINS	END	JOIN
ALL	CONTAINSTABLE	ERRLVL	KEY
ALTER	CONTINUE	ESCAPE	KILL
AND	CONVERT	FILLFACTOR	LEFT
ANY	CREATE	FOR	LIKE
AS	CROSS CURRENT	FOREIGN	LINENO
ASC	CURRENT_DATE	FREETEXT	LOAD
AUTHORIZATION	CURRENT_TIME	FREETEXTTABLE	NATIONAL
BACKUP	CURRENT_TIMESTAMP	FROM	NOCHECK
BEGIN	CURRENT_USER	FULL	NONCLUSTERED
BETWEEN	CURSOR	FUNCTION	NOT
BREAK	DATABASE	GOTO	NULL
BROWSE	DBCC	GRANT	NULLIF
BULK	DEALLOCATE	GROUP	OF
BY	DECLARE	HAVING	OFF
CASCADE	DEFAULT	HOLDLOCK	OFFSETS
CASE	DELETE	IDENTITY	ON
CHECK	DENY	IDENTITY_INSERT	OPEN
CHECKPOINT	DESC	IDENTITYCOL	OPENDATASOURCE
CLOSE	DISK	IF	OPENQUERY
CLUSTERED	DISTINCT	IN	OPENROWSET
COALESCE	DISTRIBUTED	INDEX	OPENXML
COLLATE	DOUBLE	INNER	OPTION
COLUMN	DROP	INSERT	OR
COMMIT	DUMMY	INTERSECT	ORDER
COMPUTE	DUMP	INTO	OUTER
CONSTRAINT	ELSE	IS	OVER

PERCENT	RESTRICT	SOME	UPDATE
PLAN	RETURN	STATISTICS	UPDATETEXT
PRECISION	REVOKE	SYSTEM_USER	USE
PRIMARY	RIGHT	TABLE	USER
PRINT	ROLLBACK	TEXTSIZE	VALUES
PROC	ROWCOUNT	THEN	VARYING
PROCEDURE	ROWGUIDCOL	TO	VIEW
PUBLIC	RULE	TOP	WAITFOR
RAISERROR	SAVE	TRAN	WHEN
READ	SCHEMA	TRANSACTION	WHERE
READTEXT	SELECT	TRIGGER	WHILE
RECONFIGURE	SESSION_USER	TRUNCATE	WITH
REFERENCES	SET	TSEQUAL	WRITETEXT
REPLICATION	SETUSER	UNION	
RESTORE	SHUTDOWN	UNIQUE	

ODBC Reserved Keywords (Source: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/tsqlref/ts_ra-rz_9oj7.asp. Date accessed: 6/22/2004).

The following words are reserved for use in ODBC function calls. These words do not constrain the minimum SQL grammar; however, to ensure compatibility with drivers that support the core SQL grammar, applications should avoid using these keywords. This is the current list of ODBC reserved keywords. For more information, see *Microsoft ODBC 3.0 Programmer's Reference, Volume 2, Appendix C*.

ABSOLUTE	COLLATE	DISCONNECT	IMMEDIATE
ACTION	COLLATION	DISTINCT	IN INCLUDE
ADA	COLUMN	DOMAIN	INDEX
ADD	COMMIT	DOUBLE	INDICATOR
ALL	CONNECT	DROP	INITIALLY
ALLOCATE	CONNECTION	END	INNER
ALTER	CONSTRAINT	END-EXEC	INPUT
AND	CONSTRAINTS	ESCAPE	INSENSITIVE
ANY	CONTINUE	EXCEPT	INSERT
ARE	CONVERT	EXCEPTION	INT
AS	CORRESPONDING	EXEC	INTEGER
ASC	COUNT	EXECUTE	INTERSECT
ASSERTION	CREATE	EXISTS	INTERVAL
AT	CROSS	EXTERNAL	INTO
AUTHORIZATION	CURRENT	EXTRACT	IS
AVG	CURRENT_DATE	FALSE	ISOLATION
BEGIN	CURRENT_TIME	FETCH	JOIN
BETWEEN	CURRENT_TIMESTAMP	FIRST	KEY
BIT	CURRENT_USER	FLOAT	LANGUAGE
BIT_LENGTH	CURSOR	FOR	LAST
BOTH	DATE	FOREIGN	LEADING

BY	DAY	FORTRAN	LEFT
CASCADE	DEALLOCATE	FOUND	LEVEL
CASCADED	DEC	FROM	LIKE
CASE	DECIMAL	FULL	LOCAL
CAST	DECLARE	GET	LOWER
CATALOG	DEFAULT	GLOBAL	MATCH
CHAR	DEFERRABLE	GO	MAX
CHAR_LENGTH	DEFERRED	GOTO	MIN
CHARACTER	DELETE	GRANT	MINUTE
CHARACTER_LENGTH	DESC	GROUP	MODULE
CHECK	DESCRIBE	HAVING	MONTH
CLOSE	DESCRIPTOR	HOUR	NAMES
COALESCE	DIAGNOSTICS	IDENTITY	NATIONAL
NATURAL	PRECISION	SIZE	TRIM
NCHAR	PREPARE	SMALLINT	TRUE
NEXT	PRESERVE	SOME	UNION
NO	PRIMARY	SPACE SQL	UNIQUE
NONE	PRIOR	SQLCA	UNKNOWN UPDATE
NOT	PRIVILEGES	SQLCODE	UPPER
NULL	PROCEDURE	SQLERROR	USAGE
NULLIF	PUBLIC READ	SQLSTATE	_USER
NUMERIC	REAL	SQLWARNING	USER
OCTET_LENGTH	REFERENCES	SUBSTRING	USING
OF	RELATIVE	SUM	VALUE
ON	RESTRICT	SYSTEM	VALUES
ONLY	REVOKE	TABLE	VARCHAR
OPEN	RIGHT	TEMPORARY ELSE	VARYING
OPTION	ROLLBACK	THEN	VIEW
OR	ROWS	TIME	WHEN
ORDER	SCHEMA	TIMESTAMP	WHENEVER
OUTER	SCROLL	TIMEZONE_HOUR	WHERE
OUTPUT	SECOND	TIMEZONE_MINUTE	WITH
OVERLAPS	SECTION	TO	WORK
PAD	SELECT	TRAILING	WRITE
PARTIAL	SESSION	TRANSACTION	YEAR
PASCAL	SESSION_USER	TRANSLATE	ZONE
POSITION	SET	TRANSLATION	

Oracle Reserved Words and Keywords (Source: http://www.nvc.cs.vt.edu/pkgdocs/Oracle/server.805/a58234/vol2_wor.htm#421705. Date accessed 6/22/2004).

Oracle reserved words have a special meaning to Oracle and so cannot be redefined. For this reason, you cannot use them to name database objects such as columns, tables, or indexes.

Keywords also have a special meaning to Oracle but are not reserved words and so can be redefined. However, some might eventually become reserved words, so care should be taken when using them as variable or function names in an application.

|	ARRAY	CHECKPOINT	CURRENT_SCHEMA
&	ARRAYLEN	CHOOSE	CURRENT_USER
:	AS	CHUNK	CURSOR
,	ASC	CLEAR	CYCLE
-	AT	CLOB	
=	AUDIT	CLONE	DANGLING
>	AUTHENTICATED	CLOSE	DATABASE
[	AUTHORIZATION	CLOSE_CACHED_OPEN_CURSORS	DATAFILE
<	AUTOEXTEND	CLUSTER	DATAFILES
(	AUTOMATIC	COALESCE	DATAOBJNO
.	AVG	COBOL	DATE
+		COLUMN	DBA
]	BACKUP	COLUMNS	DEALLOCATE
)	BECOME	COMMENT	DEBUG
!	BEFORE	COMMIT	DEC
/	BEGIN	COMMITTED	DECIMAL
*	BETWEEN	COMPATIBILITY	DECLARE
^	BFILE	COMPILE	DEFAULT
@	BITMAP	COMPLETE	DEFERRABLE
	BLOB	COMPOSITE_LIMIT	DEFERRED
ACCESS	BLOCK	COMPRESS	DEGREE
ACCOUNT	BODY	COMPUTE	DELETE
ACTIVATE	BY	CONNECT	DEREF
ADD		CONNECT_TIME	DESC
ADMIN	CACHE	CONSTRAINT	DIRECTORY
ADVISE	CACHE_INSTANCES	CONSTRAINTS	DISABLE
AFTER	CANCEL	CONTENTS	DISCONNECT
ALL	CASCADE	CONTINUE	DISMOUNT
ALL_ROWS	CAST	CONTROLFILE	DISTINCT
ALLOCATE	CFILE	CONVERT	DISTRIBUTED
ALTER	CHAINED	COST	DML
ANALYZE	CHANGE	COUNT	DOUBLE
AND	CHAR	CPU_PER_CALL	DROP
ANY	CHAR_CS	CPU_PER_SESSION	DUMP
ARCHIVE	CHARACTER	CREATE	EACH
ARCHIVELOG	CHECK	CURRENT	ELSE

ENABLE	GRANT	LIBRARY	NATIONAL
END	GROUP	LIKE	NCHAR
ENFORCE	GROUPS	LIMIT	NCHAR_CS
ENTRY		LINK	NCLOB
ESCAPE	HASH	LIST	NEEDED
ESTIMATE	HASHKEYS	LISTS	NESTED
EVENTS	HAVING	LOB	NETWORK
EXCEPT	HEADER	LOCAL	NEW
EXCEPTIONS	HEAP	LOCK	NEXT
EXCHANGE	IDENTIFIED	LOCKED	NOARCHIVELOG
EXCLUDING	IDGENERATORS	LOG	NOAUDIT
EXCLUSIVE	IDLE_TIME	LOGFILE	NOCACHE
EXEC	IF	LOGGING	NOCOMPRESS
EXECUTE	IMMEDIATE	LOGICAL_READS_PER_CALL	NOCYCLE
EXISTS	IN	LOGICAL_READS_PER_SESSION	NOFORCE
EXPIRE	INCLUDING	LONG	NOLOGGING
EXPLAIN	INCREMENT		NOMAXVALUE
EXTENT	INDEX	MANAGE	NOMINVALUE
EXTENTS	INDEXED	MANUAL	NONE
EXTERNALLY	INDEXES	MASTER	NOORDER
	INDICATOR	MAX	NOOVERRIDE
FAILED_LOGIN_ATTEMPTS	IND_PARTITION	MAXARCHLOGS	NOPARALLEL
FALSE	INITIAL	MAXDATAFILES	NORESETLOGS
FAST	INITIALLY	MAXEXTENTS	NOREVERSE
FETCH	INITRANS	MAXINSTANCES	NORMAL
FILE	INSERT	MAXLOGFILES	NOSORT
FIRST_ROWS	INSTANCE	MAXLOGHISTORY	NOT
FLAGGER	INSTANCES	MAXLOGMEMBERS	NOTFOUND
FLOAT	INSTEAD	MAXSIZE	NOTHING
FLOB	INT	MAXTRANS	NOWAIT
FLUSH	INTEGER	MAXVALUE	NULL
FOR	INTERMEDIATE	MIN	NUMBER
FORCE	INTERSECT	MEMBER	NUMERIC
FOREIGN	INTO	MINIMUM	NVARCHAR2
FORTRAN	IS	MINEXTENTS	
FOUND	ISOLATION	MINUS	OBJECT
FREELIST	ISOLATION_LEVEL	MINVALUE	OBJNO
FREELISTS	KEEP	MLSLABEL	OBJNO_REUSE
FROM	KEY	MODE	OF
FULL	KILL	MODIFY	OFF
FUNCTION		MODULE	OFFLINE
GLOBAL	LABEL	MOUNT	OID
GLOBALLY	LANGUAGE	MOVE	OIDINDEX
GLOBAL_NAME	LAYER	MTS_DISPATCHERS	OLD
GO	LESS	MULTISET	ON
GOTO	LEVEL		ONLINE

ONLY	QUEUE	SD_SHOW	SYSTEM
OPCODE	QUOTA	SECTION	
OPEN	RANGE	SEGMENT	TABLE
OPTIMAL	RAW	SEG_BLOCK	TABLES
OPTIMIZER_GOAL	RBA	SEG_FILE	TABLESPACE
OPTION	READ	SELECT	TABLESPACE_NO
OR	REAL	SEQUENCE	TABNO
ORDER	REBUILD	SERIALIZABLE	TEMPORARY
ORGANIZATION	RECOVER	SESSION	THAN
OVERFLOW	RECOVERABLE	SESSION_CACHED_CURSORS	THE
OWN	RECOVERY	SESSIONS_PER_USER	THEN
	REF	SET	THREAD
PACKAGE	REFERENCES	SHARE	TIMESTAMP
PACKED	REFERENCING	SHARED	TIME
PARALLEL	REFRESH	SHARED_POOL	TO
PARTITION	RENAME	SHRINK	TOPLEVEL
PASSWORD	REPLACE	SIZE	TRACE
PASSWORD_GRACE_TIME	RESET	SKIP	TRACING
PASSWORD_LIFE_TIME	RESETLOGS	SKIP_UNUSABLE_INDEXES	TRANSACTION
PASSWORD_LOCK_TIME	RESIZE	SMALLINT	TRANSITIONAL
PASSWORD_REUSE_MAX	RESOURCE	SNAPSHOT	TRIGGER
PASSWORD_REUSE_TIME	RESTRICTED	SOME	TRIGGERS
PASSWORD_VERIFY_FUNCTION	RETURN	SORT	TRUE
PCTFREE	RETURNING	SPECIFICATION	TRUNCATE
PCTINCREASE	REUSE	SPLIT	TX
PCTTHRESHOLD	REVERSE	SQL	TYPE
PCTUSED	REVOKE	SQLBUF	
PCTVERSION	ROLE	SQLCODE	UB2
PERCENT	ROLES	SQLERROR	UBA
PERMANENT	ROLLBACK	SQLSTATE	UID
PLAN	ROW	SQL_TRACE	UNARCHIVED
PLI	ROWID	STANDBY	UNDER
PLSQL_DEBUG	ROWLABEL	START	UNDO
POST_TRANSACTION	ROWNUM	STATEMENT_ID	UNION
PRECISION	ROWS	STATISTICS	UNIQUE
PRESERVE	RULE	STOP	UNLIMITED
PRIMARY		STORAGE	UNLOCK
PRIOR	SAMPLE	STORE	UNPACKED
PRIVATE	SAVEPOINT	STRUCTURE	UNRECOVERABLE
PRIVATE_SGA	SB4	SUCCESSFUL	UNTIL
PRIVILEGE	SCAN_INSTANCES	SUM	UNUSABLE
PRIVILEGES	SCHEMA	SWITCH	UNUSED
PROCEDURE	SCN	SYS_OP_ENFORCE_NOT_NULL$	UPDATABLE
PROFILE	SCOPE	SYSDATE	UPDATE
PUBLIC	SD_ALL	SYSDBA	USAGE
PURGE	SD_INHIBIT	SYSOPER	USE

USER	VALUE	VIEW WHEN	WORK
USING	VALUES		WRITE
	VARCHAR	WHENEVER	
VALIDATE	VARCHAR2	WHERE WITH	XID
VALIDATION	VARYING	WITHOUT	